云用户端软件 VINCA ProUsage

王菁 丁维龙 赵栓 李锋

摘要 随着基于开放网络环境的云基础设施的建立，以及 XaaS 的服务提供模式的发展，最终用户可以“使用而不拥有”、“按需”、“泛在”的方式消费和利用云环境中的资源。然而，在享受集中式的规模经济所带来的好处的同时，用户自主控制的需求也愈发突显。如何让用户能有效利用云基础设施，同时也拥有个性化应用定制、多样化终端利用、私有数据保护等自主控制能力，成为云用户端研究的关键问题之一。本文首先概述云用户端的相关工作，然后介绍我们研发的 VINCA ProUsage——一种新型云用户端软件的构成和原理。

关键词 云用户端 用户主导 最终用户编程 本地资源集成

1 引言

在跨域信息共享和应用协同的需求驱动下，以内容为中心的云“资源中心”日益受到重视。通过建立基于互联网的信息处理基础设施，可为上层应用提供计算、存储、数据和软件等资源的逻辑中心，开放地纳入网络资源，以及优化管理和调度网络资源，并以 XaaS 的模式对外提供服务。这种模式下，用户以“使用而不拥有”、“按需”、“泛在”的方式消费和利用云环境中的资源[1]。用户不需要自己构建 IT 基础设施和部署软件，而是由专业的运营商负责进行运维管理，用户以“即用即付（Pay as you go）”的方式按需使用。与传统模式相比，可以大大减少用户的前期资金投入和风险，降低管理成本，以及提高业务灵活性。从运营商的角度来看，这种集中运维的模式可有效利用规模经济来降低平均成本。云计算是典型的规模经济驱动的产物。运营商通过对资源的集中管理和调度优化，尽可能地共享资源，提高整体利用率，从而实现成本和能耗的降低。显然，这种集中式的管理运维方式可以为用户和运营商带来双赢。

然而，在享受集中式的规模经济所带来的好处的同时，用户自主控制的需求也愈发突显，主要包括以下几个方面：

■ 用户需求的个性化：

云计算环境的开放性以及规模巨大、多样的用户群体为应用构造带来了挑战，传统依赖 IT 专业人员按完备的规格需求编程构建系统的模式已经难以满足庞大的互联网用户群体自发、动态、即时的应用需求。应用构造模式逐渐从由 IT 人员分析需求构建应用的模式向以最终用户为中心按需动态构造应用的模式转变。用户需要能以按需定制或配置等用户端编程的手段来构造个性化的、能满足其即时需求的应用。

■ 终端的多样化：

终端是用户访问云计算环境的入口。随着硬件技术和移动互联网的发展，终端不再局限于传统的 PC 机，还包括手机、平板电脑、电视机等多种类型。丰富的终端设备可支持用户不受时间和地点的限制，随时随地地访问云环境中的资源。用户的使用环境也不再只是传统

的网页浏览器，还包括可综合利用云端和用户端资源的“富客户端[1]”软件，结合终端高效、友好的用户交互以及云环境的强大处理能力，为用户提供更好的使用体验。

■ 私有数据的保护：

用户一方面希望利用云环境中的服务来减少自己购买、开发和维护的成本和风险，另一方面又由于政策、法律和知识产权等因素，希望仍然将某些核心数据和业务处理放在用户端，保留对核心业务的自主控制，避免将私有数据传送到网络中所导致的私有资源泄漏危险。

针对上述要求，本文首先概述当前研究工作的进展，然后介绍 VINCA ProUsage—我们研发的一种新型云用户端软件的原理和关键技术。

2 云用户端研究概述

结合上文中讨论的云用户端需要解决的三方面问题，本章分别从最终用户编程、一体化终端利用以及私有数据保护三个角度概述相关研究工作。

2.1 最终用户编程

最终用户编程[2]是为非专业软件开发人员提供一套技术以让他们能够在一定程度上创建或修改软件产品[2,3]。随着服务计算的发展，最终用户编程技术的应用领域也从早期的桌面软件发展到云计算环境下的基于服务组合的云应用。服务作为面向领域的可复用组件，相比对象、组件形式的可复用零部件，更能直接反映用户业务需求，并且实现技术也具有跨平台、跨编程语言的特点。因此服务形式的可复用零部件更易于实现集成。最终用户主导的服务组合可支持最终用户自主按需组装所处领域内可复用的服务组件，得到可满足其个性化需求的软件实体。在服务环境下，一方面用户可以在一组粒度更大、更贴近用户需求的组件上进行组合编程，减少了编程的复杂性；另一方面服务的“使用而不拥有”的特性也为用户提供了更开放的选择范围，同时避免了用户对服务进行维护升级等负担。

按照用户关注点的不同，最终用户主导的服务组合可分为面向数据、面向过程和面向界面三类。

面向数据的最终用户服务组合是指用户通过组合不同的提供数据的服务来得到所需要的数据。在这类服务组合过程中，往往会为用户在界面上直接呈现其所感兴趣的数据，用户通过对数据进行操作来得到显式的结果。这种方式也被称为直接数据编程[3]。其所需的关键技术有：面向用户的数据模型、数据操作、全局模型与局部模型的映射和变更维护等。根据数据组织呈现方式的不同，此类工作可分为基于树形结构的数据服务、基于电子二维表格（Spreadsheet）结构的数据服务组合，以及基于嵌套表格结构的数据服务组合。典型工作包括 MashMaker[4]，SheetMusiq[5]，AMICO[6]，SpreadATOR[7]等。

在面向过程的最终用户服务组合中，用户通过构造流程的方式描述组合逻辑来组合不同功能的服务。此类工作往往为用户提供可视化的构造环境。用户可以直接通过拖拽、关联等操作来完成流程的构造。这种方式可以弥补面向数据的服务组合的缺点。在面向数据的服务组合中，程序逻辑由系统通过对用户行为的记录和推导生成，用户并不了解程序逻辑，难以发现系统推导过程中可能存在的缺陷。而在面向过程的服务组合中，为用户直接呈现的是通

[1] Rich client
[2] end user programming，EUP
[3] data-direct programming

过可视化的方式表达的程序逻辑，方便用户观察和修改程序。此类工作可分为基于数据流的服务组合和基于控制流的服务组合两类。典型工作包括 Yahoo Pipes[8]，IBM Damia[9]以及 Marmite[10]等。

用户界面是用户与计算机操作的接口。面向界面的最终用户服务组合旨在支持最终用户通过界面层的组合来设计和创建应用。对于用户来说，所见到的是界面组件，服务的接口信息被隐藏在呈现层之后。这种方式可以为用户提供“所见即所得”的用户体验。此类工作中的关键技术包括服务组件的界面生成、界面交互以及遗留应用界面集成等。此类工作可分为客户端界面组件的集成、浏览器界面组件的集成和 Web 应用的界面集成几类。典型工作包括 FAST Gadgets Visual Storyboard[11]，ServFace Builder [12]等。

2.2 一体化终端利用

纵观网络计算的发展历史，可以发现用户使用环境的架构在客户端和服务器端之间经过了多次反复、螺旋式渐进。早期的计算机系统一般体积庞大、造价昂贵，资源和服务都集中在大型主机上，因此必须采用瘦客户端。用户通过 3270 等终端机登录主机，进行相关的业务操作。从上世纪 80 年代中期开始，传统的大型主机与哑终端受到了以 PC 为主体的微机网络的挑战，客户端/服务器（C/S）逐渐成为主流模式。客户端应用程序可以为用户提供丰富的界面元素和快速的本地运算能力。随着互联网的发展，带宽逐渐提高，瘦客户端模式重新逐渐得到发展。其中典型代表有网络计算机（NC）和浏览器/服务器(B/S)架构等。瘦客户端模式在减少对客户端的要求，降低管理维护成本的同时，也增加了对服务器的依赖，具有必须在线工作、网络传输量大、服务器端容易形成单点失效或瓶颈等缺点。随着计算机硬件技术与新一代软件技术的发展，客户端计算能力的利用再度得到了重视。在网络计算架构方面，出现了将原本在服务器端的处理任务向客户端转移的趋势。当前的一种典型模式是取消客户端和服务器的界限，即对等网络[4]的计算模式；另一种典型模式是重新划分客户端和服务器端的处理任务，千兆信息集团（Giga Information Group）称之为“回归到富终端”（Return to Rich Client），高德纳集团（Gartner Group）则将之称为高保真度客户端（High fidelity client）或是智能客户端（Smart client）。这种富客户端模式是针对传统浏览器/服务器架构的缺点而提出，重新定义了呈现层、业务逻辑层和数据层的部署及通讯方式，将一部分原本在服务器端的处理任务转移到客户端进行，这种模式下构造的应用程序被称为 RIA(Rich Internet Application[5])，其代表技术有 AJAX, Flex, JavaFX, Smart Client, SilverLight 等。

随着富客户端技术的发展，以及终端设备的种类日渐丰富、处理能力不断提升，用户的使用环境不再只是传统的网页浏览器，而演化为可综合利用云端和客户端资源的 RIA 型应用[13]。客户端的作用不仅是接收用户输入和显示页面，还包括根据用户请求与服务器端进行异步通讯、在本地维护数据，以及支持离线处理及数据同步等能力。这一切都可以在不依赖客户端与服务器端之间持续网络连接的情况下进行，因此不仅为用户带来了更好的交互体验，还尤其适合于在移动互联网环境下的各类移动终端设备上使用。

根据主控端的不同，RIA 型应用可分为服务器端为控制中心以及客户端为控制中心两大类。

在服务器端为控制中心的情况下，用户本地的数据和程序需要注册到服务器端，在服务器端进行统一管理和控制。为了支持客户端资源利用以及网络环境变化下的离线处理，需要有客户端数据缓存、网络监测及离线/在线自动切换、数据同步及一致性保障、客户端程序

4 peer to peer，P2P
5 有人译成“富互联网应用”或“多样化互联网应用”

自动安装及更新维护机制等关键技术的支撑。典型工作有虚拟化桌面领域的 Mokafive[14]，该产品提供了嵌入开源的 VMware Player 的“桌面即服务”平台，服务器集中存储虚拟机镜像，用户可以离线或在线运行镜像，镜像可通过 Web 登录使用或下载使用，也可预装在 USB 设备、手机或者手提电脑上，并可支持虚拟机的无缝升级和自动安装卸载。也有一些工作支持基于浏览器的离线 Web 应用，如 Google Calendar, Docs 和 Gmail 等通过利用 Google Gears 技术[15]，Web 应用程序可通过 Java 脚本应用程序接口（JavaScript API）存取客户端本地的 SQLite 数据库，使得对数据的操作直接在本地进行，为用户提供更快的响应速度和更好的离线支持。

在客户端为控制中心的情况下，应用的管理和控制在用户的本地环境完成。运行时，应用被分解为可分布执行的应用片段，分别分配到本地环境或网络环境中执行。除了与服务器端为控制中心的工作所共同关注的客户端离线使用技术外，此类工作还重点关注应用划分策略、性能监测和预测技术及优化调度机制等。根据应用划分的时间，相关工作可分为静态划分和动态划分两类。前者如杨凡（音译，Fan Yang）[16]研究了数据驱动的 Web 应用的划分方法，以优化用户响应时间；后者如英特尔实验室研发的 CloneCloud[17]，可综合考虑设备处理能力、网络、云服务以及工作负载来计算划分方案，并且根据运行时的环境变化动态自适应调整。

2.3 私有数据保护

在云计算环境下，数据的隐私保护面临非常严峻的考验。由于在云计算环境下用户的数据是在运营商的机器上存储和处理，用户不能够自主控制，因此，隐私问题是影响用户采纳云计算方案时要考虑的主要因素[18]。由于政策、法律、知识产权等因素，用户往往希望将某些核心数据和业务处理放在用户端，保留对核心业务的自主控制，以避免由于将私有数据传送到网络环境中而可能导致的私有资源泄漏的危险。

当前私有性保障的相关研究中，根据操作的对象，可划分面向服务资源和面向数据资源两类。面向服务资源的私有性保护是指根据服务的信任级别来选取调度资源，如 CCOF[19]、GridSec[20]等。面向数据资源的私有性保护可根据是否考虑数据资源之间的关系，分为原子私有性和约束私有性两类。

原子私有性不考虑数据资源之间的关系，对单个原子数据资源进行私有性策略描述和保护，较容易实现。但其实现对用户提出了较高的要求，需要用户自行描述每个原子资源的私有性策略，保证策略的一致性。原子私有性保护的相关技术分为基于信息隐藏的数据保护技术和基于策略的数据保护技术两类。基于信息隐藏的数据保护通过密码、哈希或者匿名技术对数据进行处理，属于在数据发送的前提下的私有性保护。基于策略的数据保护通过在数据发送前对请求和资源进行匹配，判断是否允许将私有数据发送给资源。例如 W3C 的 P3P(Platform for Privacy Preferences) [21]，用于判断 Web 站点的隐私保护策略是否符合用户的偏好，保护的对象是用户浏览网页相关的个人信息。文献[22]提出了情形敏感的访问控制机制，在服务匹配时根据情形决定提供哪些信息，对服务请求的信息是否可见进行了控制。

约束私有性考虑数据资源之间的关系，在策略模型中包含了对数据关系的描述，以及数据关系对私有性的约束，可以简化用户的私有性策略表达，以及提供更丰富的约束需求。约束私有性保护的相关技术有：支持数据关系描述的私有性策略模型、策略一致性判定以及保持策略一致性的应用调度机制。文献[23]考虑了数据依赖关系，提出了一个私有性保护的服务组合框架，通过对应用数据采用敏感度进行标注，并和服务模型进行匹配来判定是否会违背私有性策略。该工作通过服务模型描述服务的输入输出参数之间的依赖关系，在进行服务

匹配时通过数据依赖将策略进行传播。文献[24]考虑了分布式工作流执行过程中竞争组织之间的利益冲突问题，提出了一种分布式工作流“中国墙”[6]模型，保护了敏感的工作流控制数据，提供安全可信的工作流分布式执行。该工作考虑了工作流控制数据对工作流执行流向的影响，限制了存在利益冲突的组织对工作流控制数据的访问。文献[25]考虑到了职责分离问题，将具有相似功能的服务进行分组，以最大限度地使职责分离作为目标进行服务调度。职责分离问题将角色划分为互斥集合，当一个用户被分配了一个角色后，不能拥有与之互斥的其它角色。与此工作类似的还有文献[26]等。

3 VINCA ProUsage 介绍

VINCA ProUsage 为用户提供了方便、易用的云用户端编程和使用环境。其突出特点是以用户为中心，一方面可连接 VINCA 服务社区，为用户提供一体化地访问和使用云环境中的资源和用户端本地资源的窗口，另一方面还可支持用户灵活地组合资源来构造个性化的可满足即时需求的应用，从而降低成本，提高应用开发效率，在使能用户对云计算基础设施的有效利用的同时，还能拥有用户端的主控能力。

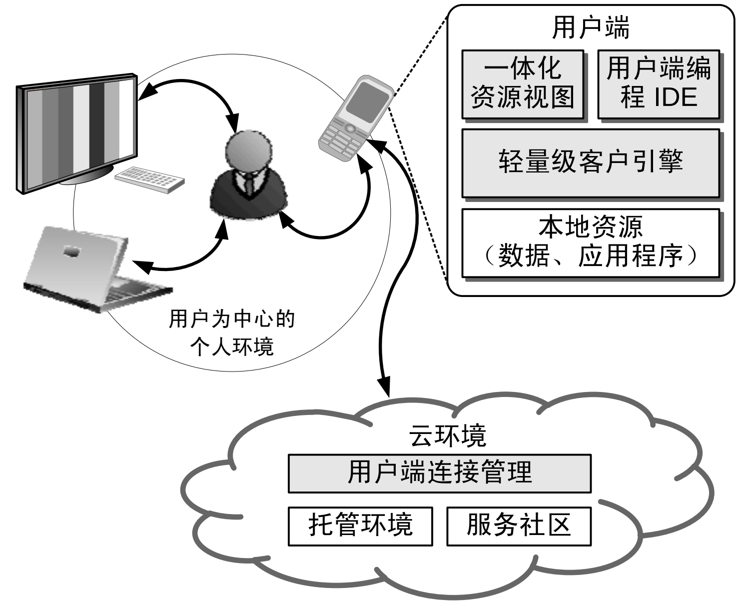


图 1. VINCA ProUsage 构成

VINCA ProUsage 主要包括用户端和云端两大部分。用户端的本地环境包括一体化资源视图、用户端编程集成开发环境[7]和轻量级客户端引擎三个模块。云端为部署在服务社区和托管环境基础上的用户端连接管理模块。如图 1 中灰色方块所示。

各模块的主要功能为：

1. **一体化资源视图**：实现与服务社区的交互以及服务资源视图在用户端的呈现，可以同时连接多个服务社区，并一体化呈现远程资源和用户端本地资源。用户可透明使用远程资源和本地资源，而无需关心资源的物理部署、调用接口等实现层面的细节。允许用户定制其所属的服务社区中感兴趣的服务资源，形成用户自主定制的个性化资源视图，为应用构造奠定基础。

2. **用户端编程集成开发环境**：提供用户主导的便捷、灵活的可视化应用集成开发环境，支持用户自主定制个性化应用，以及对应用进行随需调整；提供应用正确性保障机制，可对用户的编程操作进行即时检验，并智能推荐与当前情景相匹配的服务资源。

[6] 即 Chinese Wall，又称防火墙、信息隔离墙，最初是在 1929 经济危机后引入的防止投资人利益冲突的一种机制。后来其意义推广为通过行政规定和划分、内部监控、IT 系统等手段防止各部门之间出现有损客户利益的利益冲突事件的机制。
[7] Integrated Development Environment, IDE

3. **轻量级客户端引擎：**将客户端作为应用执行的控制中心来组织和调度资源。这种模式可以看作是为用户构建了一种安全透明的应用执行环境，可对应用进行自动分解、调度、协调和合成，在优化使用资源的同时，对本地资源进行隔离保护。
4. **用户端连接管理：**实现用户端与服务器端的连接建立、会话维护和迁移、执行请求提交与执行结果接收等交互功能，以支持通过多样化终端设备、以及在多种网络环境下无缝地连接和使用服务器端的资源。

VINCA ProUsage 的技术特色为：

1. 用户主导的业务应用构造：

以支持最终用户编程为目标，VINCA ProUsage 可支持面向过程和面向数据两类不同的服务组合应用需求，提出了探索式的服务编排、基于业务服务的业务流程建模和基于嵌套表格的 Mashup 应用构造三种不同的最终用户服务组合的技术。其中，前两者均从面向过程角度出发，探索式服务编排主要针对动态和即时的应用需求，提供一种“边定义、边执行”的柔性过程建模方式；基于业务服务的业务流程建模则主要针对典型和稳定的业务流程管理需求，提供一种业务用户可自主定义完成可执行业务流程的建模方式。

在用户编程的系统辅助支撑层面，VINCA ProUsage 提供了情景敏感的服务主动推荐和自动关联机制，可为用户提供智能化建议，实现资源的“随需而现”，帮助用户决策；提供了即时验证机制，可检验和预测用户构建的应用是否符合系统中已有的约束（矛盾与否和相容性等），并对不一致的程序进行度量，向用户反馈验证结果和调整建议，提高用户编程的质量和效率。

2. 客户端为中心的应用调度执行：

VINCA ProUsage 采用了一种以客户端为中心的计算模型。客户端作为应用调度和执行的控制中心，组织和使用本地及网络资源，并协调客户端和远程节点的运行。轻量级客户端引擎在综合考虑网络状况、客户端和服务器能力等情况下，进行业务应用的分解、调度、协调执行和数据合成等操作。客户端作为控制中心主动参与到应用的执行，而不是被动地依赖服务器的工作。客户端引擎的一个重要功能是对应用的重调度。当服务状态或者网络环境发生变化时，客户端引擎根据重调度策略实时、动态地对尚未执行的应用片段进行调度和部署，以保证应用的不中断执行，并能获得最优的执行效率。

3. 私有数据保护：

VINCA ProUsage 提供了用户端私有数据的保护机制，可支持用户描述单项数据资源的私有性策略以及数据之间的依赖、互斥等约束关系。对于单个数据资源的允许或禁止公开的策略可以在应用运行前就进行判断；而通过约束关系描述的策略往往与应用运行时的信息相关，需要在运行时进行判断。系统可对策略集进行一致性验证和监控，通过静态分析和动态调整相结合的应用调度机制，不仅可检测应用设计时的私有资源泄漏危险，还可在运行时判断是否会有冲突发生，以及对冲突进行及时调解，保障了应用不会违背用户设定的保护本地数据的私有性策略。通过私有数据保护机制，用户私有的数据、程序等可以在本地环境中直接使用，无须包装成服务、注册到网络环境中的注册中心的繁琐步骤，并且避免了私有资源泄露的问题。

4 小结

随着云计算基础设施和终端设备的发展，计算本身正逐渐由“以机器为中心”向“以用户为中心”转变。为了支持最终用户充分利用云基础设施，云用户端的研究得到了工业界和学术界的普遍关注。本文分析了云用户端的自主控制问题，将其分解为支持用户的个性化需求、多样化终端利用以及私有数据保护三方面，讨论了当前的研究工作进展情况，然后介绍了云用户端软件 VINCA ProUsage 的研发。VINCA ProUsage 为用户提供了方便、易用的云用户端编程和使用环境，能支持用户主导的业务应用构造、客户端为中心的应用调度执行以及私有数据保护，可兼顾对云计算基础设施的有效利用和用户端的主控能力，提供更好的用户使用体验，有助于促进云计算的广泛推广和应用。

参考文献：

[1] 韩燕波，王桂玲，刘晨等，互联网计算的原理与实践——探索网格、云和 Web X.0 背后的本质问题和关键技术．北京：科学出版社, 2010.

[2] Eisenberg M. End-user programming handbook of human-computer interaction. Amsterdam: Elsevier Science. 1997.

[3] Jones C. End user programming. Computer, 1995.28(9): 68~70.

[4] Ennals R, Gay D. User-friendly functional programming for web mashups Proceedings of the 12th ACM SIGPLAN international conference on Functional programming: 2007. 223~234.

[5] Liu, B. and H. Jagadish. A spreadsheet algebra for a direct data manipulation query interface. in VLDB. 2009: IEEE Computer Society Washington, DC, USA.

[6] Obrenović, Ž. and D. Gašević, End-User Service Computing: Spreadsheets as a Service Composition Tool. IEEE Transactions on Services Computing, 2008. 1: p. 229-242.

[7] Kongdenfha, W., et al. Rapid development of spreadsheet-based web mashups. in Proceedings of the 18th international conference on World wide web. 2009. Madrid, Spain: ACM.

[8] Jones, M.C., E.F. Churchill and M.B. Twidale. Mashing up visual languages and web mash-ups. in Visual Languages and Human-Centric Computing, 2008. VL/HCC 2008. IEEE Symposium on. 2008.

[9] Altinel, M., et al. Damia: a data mashup fabric for intranet applications. in Proceedings of the 33rd international conference on Very large data bases. 2007. Vienna, Austria: VLDB Endowment.

[10] Wong, J. and J.I. Hong. Making mashups with marmite: towards end-user programming for the web. in Proceedings of the SIGCHI conference on Human factors in computing systems. 2007. San Jose, California, USA: ACM.

[11] V. Hoyer, T. Janner, I. Delchev, et al., "The FAST Platform: An Open and Semantically-Enriched Platform for Designing Multi-channel and Enterprise-Class Gadgets," in Service-Oriented Computing. vol. 5900, Springer Berlin / Heidelberg, 2009, pp. 316-330.

[12] Marius Feldmann, Jordan Janeiro, Tobias Nestler, Gerald Hübsch, Uwe Jugel, Andre Preußner, Alexander Schill. An Integrated Approach for Creating Service-Based Interactive Applications; Conference in Human-Computer Interaction (Interact); 2009

[13] Mario Höfer, Gernot Howanitz, The Client Side of Cloud Computing, University of Salzburg Jul 1, 2009

[14] MokaFive. http://www.mokafive.com/

[15] Google Gears. http://gears.google.com/

[16] F. Yang, N. Gupta, N. Gerner, X. Qi, A. Demers, J. Gehrke, and J. Shanmugasundaram. A unified platform for data driven web applications with automatic client-server partitioning. In WWW, 2007.

[17] B.-G. Chun and P. Maniatis. Augmented Smart phone Applications through Clone Cloud Execution. In HotOS, 2009.

[18] Pearson S. 2009. Taking account of privacy when designing cloud computing services. Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing: 44~52.

[19] S. Y. Zhao and V. Lo. Result Verification and Trust-based Scheduling in Open Peer-to-Peer Cycle Sharing Systems. Technical Report, University of Oregon, USA, 2005.

[20] S. S. Song, Y. K. Kwok, and K. Hwang. Trusted Job Scheduling in Open computational Grids: Security-Driven heuristics and A Fast Genetic Algorithm. In 19th IEEE International Parallel & Distributed Processing Symposium (IPDPS-2005), Denver, CO, USA., IEEE Computer Society Press, Los Alamitos, CA, USA., April 4-8, 2005.

[21] L. Cranor. The Platform for Privacy Preferences 1.1 (P3P1.1) Specification. W3C working draft, [Online]. Available from URL: http://www.w3.org/P3P/. 2004.

[22] S. Yau and J. Liu. A Situation-aware Access Control based Privacy-Preserving Service Matchmaking Approach for Service-Oriented Architecture. 2007 IEEE International Conference on Web Services (ICWS 2007), 2007, 1056-1063.

[23] W. Xu, V.N. Venkatakrishnan, R. Sekar, and I.V. Ramakrishnan. A Framework for Building Privacy-Conscious Composite Web Services. 4th IEEE International Conference on Web Services (ICWS'06) (Application Services and Industry Track), Chicago, IL, September 2006, 655-662.

[24] Vijayalakshmi, C. Soon Ae, and M. Pietro. Chinese Wall Security for Decentralized Workflow Management Systems. Journal of Computer Security, 2004, 12(6): 799-840.

[25] P. C. K. Hung. From Conflict of Interest to Separation of Duties in WS-Policy for Web Services Matchmaking Process, In Proceedings of the IEEE Thirty-Seventh Hawaii International Conference on System Sciences (HICSS-37), Hawaii, USA, 2004, 3: 30066b.

[26] H. J. Wang, and Z. Leon. A Formal Approach to Verification of Process Constraints. 16th Annual Workshop on Information Technologies & Systems (WITS), 2007, Paper Available at SSRN: http://ssrn.com/abstract=1025600.

作者简介：

王 菁： 中国科学院计算技术研究所软件集成与服务计算研究分中心助理研究员、博士 wangjing@ict.ac.cn

丁维龙： 中国科学院计算技术研究所软件集成与服务计算研究分中心博士研究生

赵 栓： 中国科学院计算技术研究所软件集成与服务计算研究分中心硕士研究生

李 锋： 中国科学院计算技术研究所软件集成与服务计算研究分中心硕士研究生